

PROJECT DEVELOPMENT PHASE SPRINT3

Date	24 October 2022
Team ID	PNT2022TMID17788
Project Name	Smart Lender - Applicant Credibility Prediction for Loan Approval
Maximum Marks	20 Marks

APPLICATION DEVELOPMENT

```
from flask import Flask, request, render_template, url_for
import numpy as np
import pickle
import requests
```

```
app = Flask(__name__)
```

```
@app.route('/')
def home():
    return render_template('Home.html',title = 'home',val = 'namaste')
```

```
@app.route('/detailPage')
def detail():
    return render_template('Details.html')
```

```
@app.route('/showResult',methods = ['POST'])
def result():
    gender,married,depend,education,self_emp,applicant_income,co_income,loan_amount,loan_term,credit_history,property_area = [x for x in
request.form.values()]
    print("details",gender,married,depend,education,self_emp,applicant_income,co_income,loan_amount,loan_term,credit_history,property_area)
    if gender == 'male':
        gender = 1
    else:
        gender = 0

    if married == 'yes':
        married = 1
    else:
        married = 0
    if depend == '3+':
        depend = 3

    if education == 'Graduate':
        education = 0
    else:
        education = 1

    if self_emp == 'yes':
        self_emp = 1
    else:
        self_emp = 0

    applicant_income = int(applicant_income)
    #applicant_income = np.log(applicant_income)

    loan_amount = int(loan_amount)
    #loan_amount = np.log(loan_amount)
    if property_area == 'Urban':
        property_area = 2

    elif property_area == 'Rural':
        property_area = 0
    else:
        property_area = 1

    features = [[gender,married,depend,education,self_emp,applicant_income,co_income,loan_amount,loan_term,credit_history,property_area]]
    print(features)
```

```

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "a1Ri3V6f6RR0LyqdgU7fIGlc0jfJaBeZ_Cqq34bQN4v6"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"field":
[["gender", "married", "depend", "education", "self_emp", "applicant_income", "co_income", "loan_amount", "loan_term", "credit_history", "property_area
"], "values": features}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/f4c91128-dd2d-440c-85da-
ab27eb9a0944/predictions?version=2022-11-15', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
pred=response_scoring.json()
prediction=pred['predictions'][0]['values'][0][0]

print(prediction)
if(prediction == 0 ):
    return render_template('Fail.html')
else:
    return render_template('Success.html')

if __name__=='__main__':
    app.run()

```